

EXPRESS to UML for XMI

Melbourne SC4 Meeting

February 2000

David Price

dmprice@us.ibm.com

+1 843 760 4341

Agenda

- **What is OMG XMI?**
- **EXPRESS to UML Mappings**
- **EXPRESS to UML Examples**

What is OMG XMI?

Existing OMG Standards

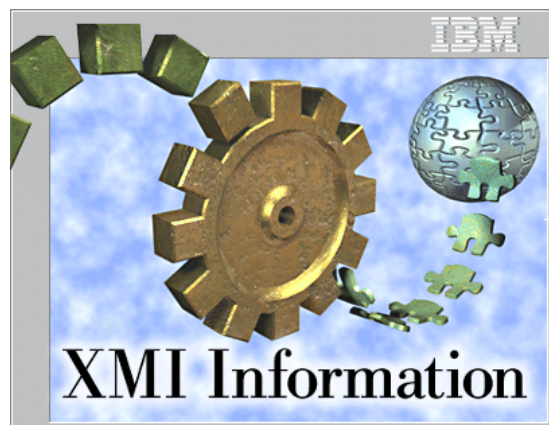
- **UML - Unified Modeling Language**
 - A graphical language for modeling systems
- **MOF - Meta-object Facility**
 - Meta-model of modeling languages
- **XMI - XML Meta-data Interchange**
 - Meta-model-based approach to the interchange of Meta-data (I.e. schemas!)
 - Can also exchange data (like Part 21) based on those schemas
- **XMI-compliant DTD for UML**
 - A lexical interchange format for UML models
 - OMG can now standardize the “UML model”, without standardizing the “UML diagram”
 - UML is no longer only graphical!

ISO
TC184/SC4

Excerpts from IBM Presentations

ISO
TC184/SC4

XMI - XML Metadata Interchange



What is XMI?

eXtensible XML Metadata Interchange

Rules and architecture for creating transfer formats

XMI 1.0 approved as OMG standard

XMI 1.1 revision under ballot now

Input: information model

Output: transfer format (DTD)

XMI Supporters

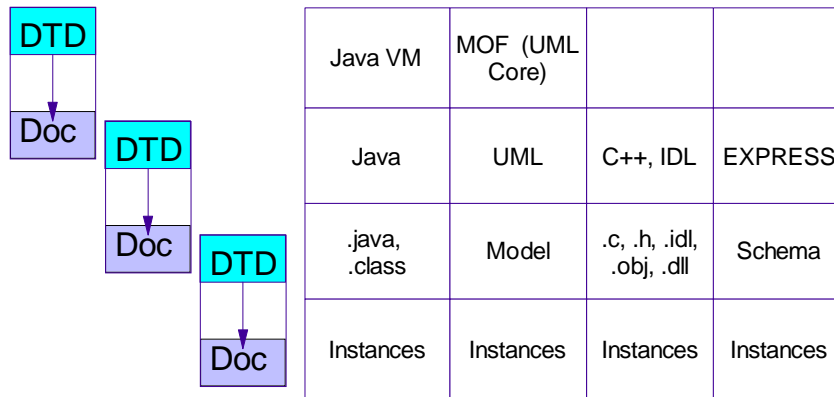
Co-Submitters (9)

- IBM, Unisys, DSTC, Oracle, Platinum, Fujitsu, Softeam, Reccerca, Daimler-Benz

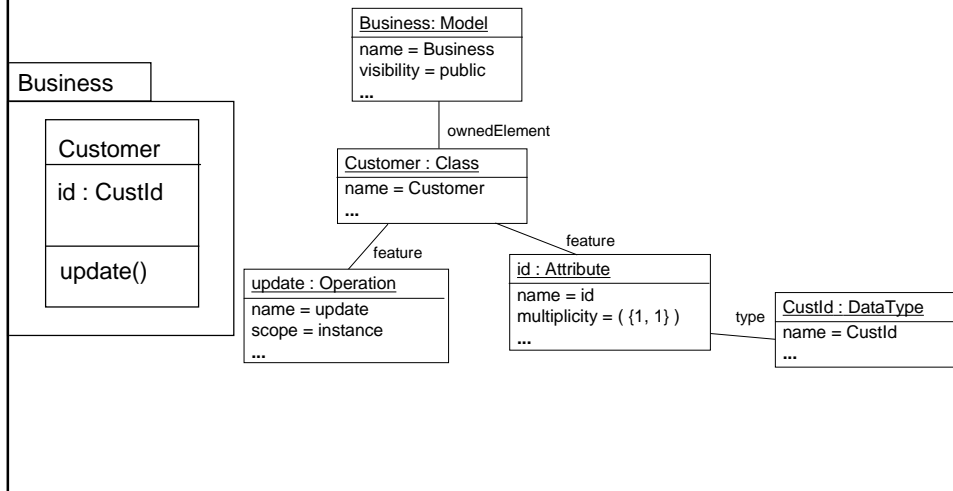
Supporters (20)

- Cayenne, Genesis, Inline, Rational, Select, Sprint, Sybase, Xerox, MCI Systemhouse, Boeing, Ardent, Aviatis, ICONIX, Integrated Systems, Verilog, Telefonica, Universitat Politecnica de Catalunya, NCR, Nihon Unisys, NTT

Common information models



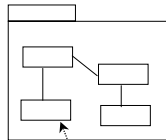
XMI Metamodel Example



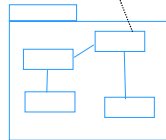
ISO
TC184/SC4

Tags from metamodel, content from model

Meta
model



Model



```
<Model>
  <name>Business</name>
  <visibility xmi.value="public"/>
  <Class>
    <name>Customer</name>
    <feature>
      <Attribute>
        <name>id</name>
        <multiplicity>
          <XMI.field>1</XMI.field>
          <XMI.field>1</XMI.field>
        </multiplicity>
      </Attribute>
    </feature>
  </Class>
</Model>
```

ISO
TC184/SC4

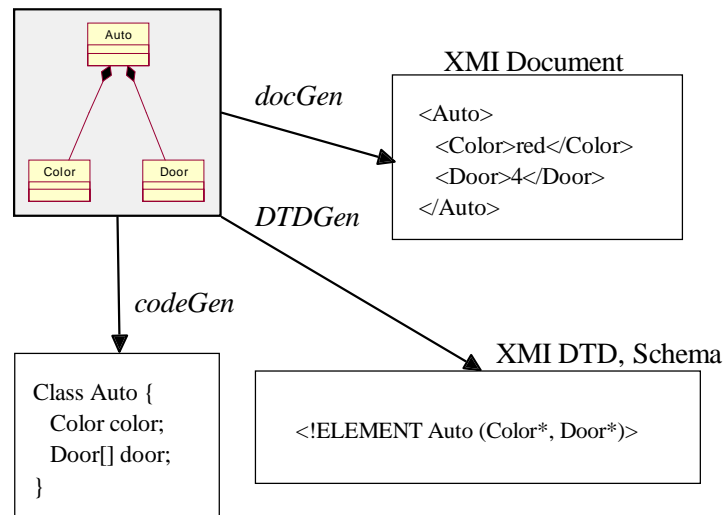
XMI: Object Interchange

Stolen by David Price
from
September '99
Steve Brodsky

sbrodsky@us.ibm.com

Design-driven XMI

Objects and Designs



Services of the XMI Toolkit

XMI Standard

- XML Open Metadata Interchange
 - ▶ IBM co-authored OMG specification with Unisys
 - ▶ Establish an industry standard specification for a stream-based model interchange format
 - ▶ Provide a generic format that can be used to transfer a wide variety of models
 - Allow exchange of OMG Object Analysis and Design Facility (OADTF) compliant models
 - CWM, UML, MOF, more coming...
- Support of 30+ vendors
- March 26, 1999 - OMG adopted technology
- XMI Revision Task Force chair - Oct '99
- UML/MOF/XMI package to be sent to ISO

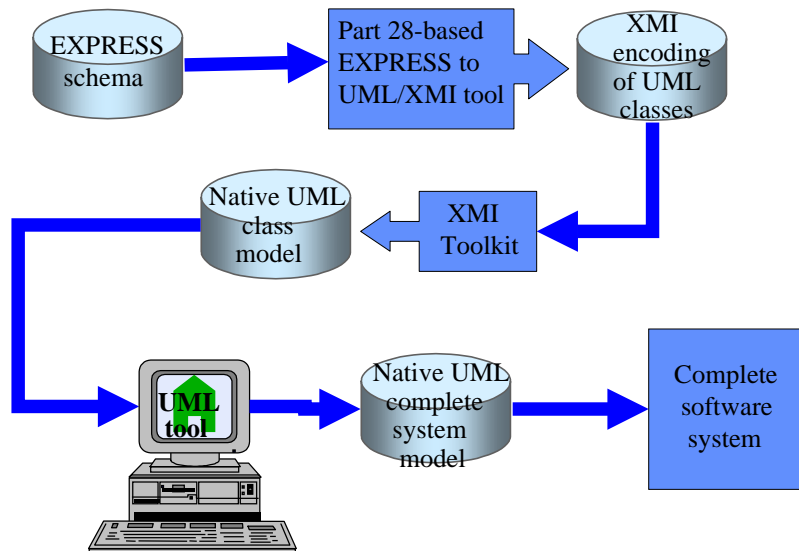
EXPRESS to UML Mappings

An Initial Proposal for Part 28

The role of schemas

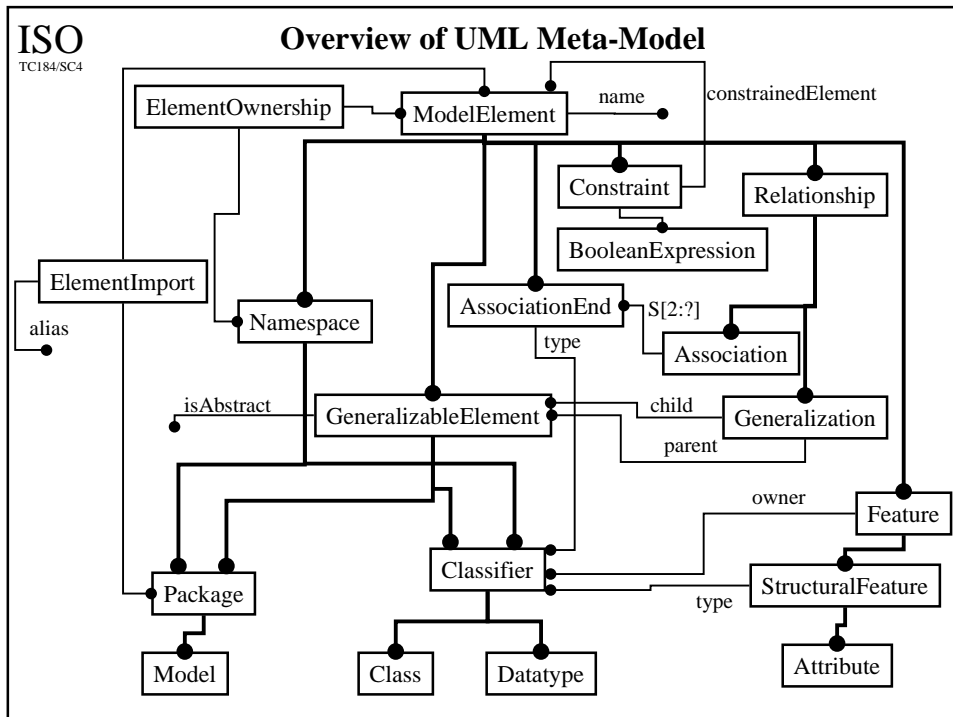
- **Main points for Part 28 project**
 - EXPRESS schema and map to XML are only a **part** of software development
 - In many scenarios, file exchange based on the EXPRESS schema never occurs
 - OMG represents a large and important community we need access to
- **Use UML as a software implementation technology, not a general purpose modeling language (I.e. a 20s, not 10s series part!)**
 - You could consider UML simply one small level of abstraction above Java or C++ classes

A Part 28/XMI Scenario



The Approach for Mapping EXPRESS to UML for XMI in Part 28

- Map to mainstream UML constructs, not EXPRESS specific extensions
- Map EXPRESS into the meta-model of UML
 - Benefit is that as XMI adds capability we do not need to change Part 28 to take advantage of it
 - For example, XMI 1.1 uses XML Namespaces and in the future XMI will produce XML Schemas as well as a DTD
- Start with simple one-way mapping, perhaps grow scope with experience
- Isolate this use of UML enable XMI use from any other UML considerations in SC4



ISO
TC184/SC4

Unmapped EXPRESS Constructs

- **FUNCTION, PROCEDURE and CONSTANT declarations not used in global or domain rules**
 - When used by a rule, these appear in a string along with the rest of the body of the rule
- **Derived attributes**
- **Remarks**

Schemas and the Interface Specification

- **One schema must be chosen as the “context” for the mapping**
- **Any EXPRESS Schema visible to “context schema” via interface specification becomes a UML Model**
- **EXPRESS elements explicitly or implicitly interfaced into context schema get mapped and are defined (via UML Element Ownership) in the UML Model representing the schema within which they were defined**
- **Each Use/Reference becomes a UML Element Import**

Aggregates

- **A one-dimensional EXPRESS aggregate becomes a UML Class (e.g. “set_of_string”)**
- **Multi-dimensional EXPRESS aggregates become a series of related UML Classes (e.g. “set_of_string” and “set_of_set_of_string”)**
 - **A UML Association links the underlying type of each aggregate with the aggregate**

Entity and subtype

- **Each EXPRESS Entity becomes a UML Class**
 - **If ABSTRACT SUPERTYPE, then UML Class “isAbstract”**
- **Each EXPRESS subtype declaration becomes a UML Generalization**
- **More complex supertype constraints are not mapped**

Defined Types

- **EXPRESS Selects become UML Classes**
 - **A UML Association links the UML Class with the other UML Classes that represent the select list**
- **EXPRESS Enumerations become UML Enumerations**
- **Other defined types become UML Classes**
 - **A UML Association links the UML Class with whatever represents the underlying type in the UML mapping**

Explicit Attributes

- **Simple or Enumeration data type-valued EXPRESS Explicit Attributes become UML Attributes**
- **All other Explicit Attributes become UML Associations**
 - **Name of Inverse Attribute is applied to Association End if declared in EXPRESS, otherwise “reverse_of_<attribute>” is created**

Constraints

- **Global rules, domain rules in ENTITY and TYPE and unique rules all map to UML Constraint that refer to whatever they constrain**
- **UML BooleanExpression contains the body of the rule in the EXPRESS language**
 - **This has two attributes: “language” and a string**
- **Functions, Procedures and Constants used in the rules are added to the end of the string representing the body of the Constraint**

EXPRESS to UML Examples

Example 4

```
SCHEMA mr_smiths_garden3;
ENTITY plant3;
END_ENTITY;
END_SCHEMA;

SCHEMA mr_jones_garden4;
USE FROM
mr_smiths_garden3(plant3)
AS mr_smiths_plant4;
END_SCHEMA;
```

```
<UML:Model id='idsmith3' name
    ='mr_smiths_garden3'>
  <UML:Class id=idplant3'
    name='plant3' />
  <UML:ElementOwnership
    namespace='idsmith3'
    ownedElement='idplant3' />
</UML:Model>

<UML:Model id='idjones4' name =
    'mr_jones_garden4'>
  <UML:ElementImport
    modelElement='idplant3'
    package='idjones4'
    visibility=public
    alias='mr_smiths_plant4' />
</UML:Model>
```

ISO
TC184/SC4

Example 5

```
SCHEMA mr_smiths_garden5;  
ENTITY garden5;  
has_bed : bed5;  
END_ENTITY;  
ENTITY bed5;  
description : STRING;  
INVERSE  
the_garden : garden5 FOR  
    has_bed;  
END_ENTITY;  
END_SCHEMA;
```

```
<UML:Model id='idsmith5' name=  
    = 'mr_smiths_garden5'>  
  <UML:Class id=idgarden5' name='garden5'>  
    <UML:Association name='has_bed'  
      type='idbed5' multiplicity='1..1' />  
    <UML:AssociationEnd name='the_garden'  
      type='idgarden5' multiplicity='1..1' />  
  </UML:Association>  
  </UML:Class>  
  <UML:ElementOwnership namespace='idsmith5'  
    ownedElement='idgarden5' />  
  <UML:Class id=idbed5' name='bed5'>  
    <UML:Attribute name='description'  
      type='idstring' />  
    <UML:Datatype id='idstring' name='string' />  
  </UML:Class>  
  <UML:ElementOwnership namespace='idsmith5'  
    ownedElement='idbed5' />  
  </UML:Model>
```

ISO
TC184/SC4

Example 8

```
SCHEMA  
mr_jones_garden8;  
ENTITY garden8;  
END_ENTITY;  
RULE only_one_garden FOR  
    (garden8);  
WHERE  
mr_jones_has_one_garden  
: SIZEOF (garden8) = 1;  
END_RULE;  
END_SCHEMA;
```

```
<UML:Model id='idjones8'  
    name='mr_jones_garden8'>  
  <UML:Class id='idgarden8' name='garden8' />  
  <UML:ElementOwnership namespace='idjones8'  
    ownedElement='idgarden8' />  
  <UML:Constraint id='idoog8'  
    name='only_one_garden' body='idboolexp8' />  
  <UML:ElementOwnership namespace='idjones8'  
    ownedElement='idoog8' />  
  <UML:Association name='constrainedElement'  
  <UML:AssociationEnd name='constrainedElement'  
    type='idgarden8' multiplicity='1..1' />  
  <UML:AssociationEnd name='constraint'  
    type='idoog8' multiplicity='1..1' />  
  </UML:Association>  
  <UML:BooleanExpression id='idboolexp8'  
    language='EXPRESS' body=  
      'WHERE  
      mr_jones_has_one_garden : SIZEOF (garden8) =  
        1;'  
  </UML:Model>
```